Multi-Class Detection and Segmentation of Objects in Depth

Cheng Zhang Hedvig Kjellström

***Abstract*— The quality of life of many people could be improved by autonomous humanoid robots in the home. To function in the human world, a humanoid household robot must be able to locate itself and perceive the environment like a human; scene perception, object detection and segmentation, and object spatial localization in 3D are fundamental capabilities for such humanoid robots. This paper presents a 3D multi-class object detection and segmentation method. The contributions are twofold. Firstly, we present a multi-class detection method, where a minimal joint codebook is learned in a principled manner. Secondly, we incorporate depth information using RGB-D imagery, which increases the robustness of the method and gives the 3D location of objects – necessary since the robot reasons in 3D space. Experiments show that the multi-class extension improves the detection efficiency with respect to the number of classes and the depth extension improves the detection robustness and give sufficient natural 3D location of the objects.**

I. Introduction

Imagine a scenario where a humanoid household robot is setting up a table for dinner: First, it needs to locate itself in kitchen. It then has to detect objects, like forks and knifes, localize and grasp them. After that it needs to detect the location of the table and what is on the table, to perform the task of putting the tableware in the right location. We can see that for a humanoid household robot to perform this kind of daily tasks, it needs to localize itself in the world, recognize and detect objects, localize them in 3D, and manipulate them.

Simultaneous localization and mapping (SLAM) is supported by scene classification (room classification) which can be done in terms of detecting objects in the room [1], [2]. Scene classification can also be employed to guide object detection and object search [3], [4]. In order to perform manipulation, detecting objects and segmenting objects in 3D serves as the preprocessing step [5].

In this paper, we present a method for *simultaneous, interleaved detection, segmentation and 3D localization of previously unseen object instances* of known categories *in unknown environments* (see Figure 1). The detection and segmentation processes guide each other in a contextual manner [6] and we exploit depth localization to constrain both detection and segmentation.

Detecting and segmenting previously unseen object instances of known classes is a long-term challenging problem. However, it is essential for many robotic applications [7], [8]. Due to its complexity, the problem is often constrained in different ways; either by detecting object *instances*, seen before [3], or by introducing attention mechanisms [9], [4]. For 3D unseen object segmentation, different amounts of knowledge about the environment is required, for example, background subtraction [10], planar support [9], environment maps [5] etc. We avoid such assumptions and perform interleaved detection and segmentation of previously unseen object instances in 3D.

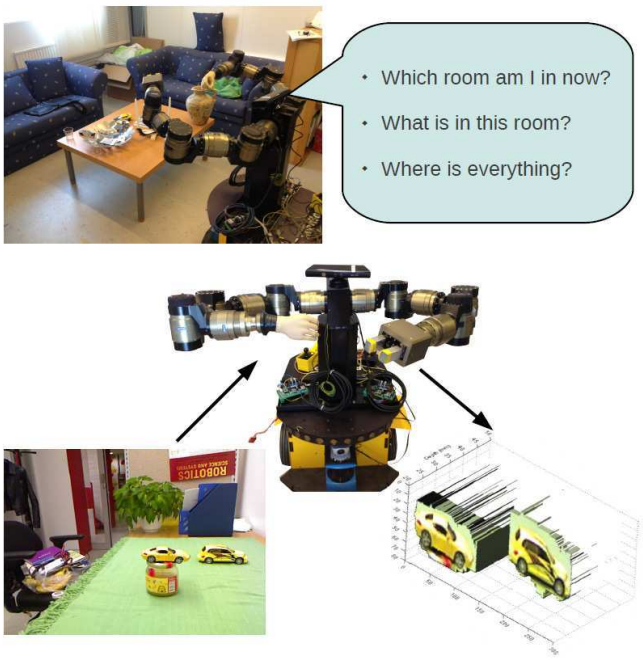


Fig. 1. An autonomous humanoid robot, which is used to do daily tasks to assist human, must be able to locate itself through scene perception, detect the target object classes which would be manipulated with, and know the objects location to guide the manipulation. This paper presents a method for interleaved detection, segmentation, and 3D location of previously unseen objects from a range of known object classes using RGB-D imagery, which can be used for robot perception.

We employ the approach of Leibe et al. [6] for interleaved object detection and segmentation, where visual words in the image are voting for the center and boundaries of regions with a specific object category in them. This method is described in Section III. The contributions of the present paper are two enhancements of the approach [6], which make the object detections applicable for robotic reasoning.

Firstly, we introduce a *multi-class detector*, which uses a joint codebook for all object classes instead of separate codebooks for each class, described in more detail in Section IV. The gain of this is in terms of efficiency: Whereas the computational cost of separate detectors for each class grows linearly with the number of object classes, we can expect

This work was supported by the Swedish Research Council.

The authors are with the Computer Vision and Active Perception Lab and the Centre for Autonomous Systems, CSC, KTH, Sweden. chengz,hedvig@kth.se

a sub-linear complexity for a multi-class detector. This is experimentally validated in Section VI-A.

Secondly, we include *depth constraints* in the voting for object centers. Depth measurements are obtained with a Kinect® camera. This increases the robustness of the detection, ruling out object hypotheses whose visual words are on significantly different depths. The underlying assumption is that the relative depths of different parts of an object are small in comparison to the distance to the camera. Furthermore, the encoded depth information also enables reasoning about object location in 3D, which is necessary for a humanoid robot application. This is further described in Section V, and experimentally evaluated in Section VI-B.

II. Related Work

The discussion of related work is divided into a review of detection work and one of segmentation.

Detection. Many object detection methods have been proposed in the past decades. The traditional approach is to use a sliding window over the whole image, accompanied by a binary classifier. Such an approach is very expensive and not scalable in terms of the number of categories. To improve efficiency, Viola and Jones [11] use a boosted cascade of successively more elaborate classifiers. Most windows could be rejected in the early stages, while more complex, slower classifiers could focus on the few difficult cases. Extensions include, unsupervised online boosting [12] and kernel methods [13].

A number of bag-of-visual-words methods have also been proposed [6], [14], [15], [16], where different ways to learn a codebook including spatial information are proposed. There is in general a trade-off between structural flexibility in the model and the ability to capture structural information: A pure bag-of-words approach can not capture structural information, but is very robust to changes in shape, articulation and view point of objects. On the other hand, a method that models the metric relationships between visual words might capture spatial structure very well, but cannot generalize over changes in scale, viewpoint and articulation.

Leibe et al. [6] (the basis of our method, described in Section III) found a good trade-off between flexibility and expressional power, using a voting based approach which enables detection of highly articulated objects in real-world scenes. Object detection and segmentation are treated as two closely collaborating processes, improving each others' results. Developed in parallel with our approach, Sun et al. [17] present work that encodes depth information in this model, enabling shape recovery in 2D and 3D, and Razavi et al. [18] propose a scalable multi-class object detection by introducing a class dimension in the voting space. However, a robot detection system must be able to handle many object classes as well as reasoning about the objects in 3D. To that end, we present a humanoid household robot perception application (developed independently from [17], [18]) which is multi-class scalable *and* encodes 3D information.

Leibe et al. use a codebook which is built from textual information. Others [14], [19], [20] build their visual words on contour-based information, with convincing results. Opelt et al. [14] present a boundary-fragment-model which uses a contour-based features with information on the location of the object's centroid. By computing votes for the object centroid, boundary fragments are selected. Shotton et al. [19] present a categorical object detection scheme that uses only local contour based features. They employ a star-based configuration which is flexible enough to cope with large variation in shape and appearance of both rigid and articulated objects. Using the star model, of Felzenszwalb et al. [15], [16] are able to detect rigid and deformable objects, using a topological grammar of object configuration.

As discussed in Section IV, the codebook size is the key factor in the computational efficiency of a codebook-based detector. The codebook size can be decreased either by removing the least informative features [21], or by making sure that classes share features [22]. Liu et al. [23] present a general probabilistic framework for codebook selection which is shown to incorporate all entropy-based measures. In this paper we use information gain, which is defined in terms of the pointwise mutual information between words and object classes. A point for future work is to formulate this in terms of the general framework of [23].

Currently, using RGB-D imagery in Robotics draws a lot of attention. Lai et al. [24] recently published a RGB-D database and used visual cues, depth cues and rough knowledge of the configuration of the setting to segment objects in video sequences. RGB and depth information are also used in *Instance Distance Learning* (IDL), proposed by Lai et al. [25]. It should be noted that IDL is only used for recognition, whereas our method addresses the more challenging problem of object detection in natural images. Furthermore, there are differences in the way depth features and visual features are integrated.

Segmentation. It is argued that segmentation plays a fundamental role in human perception [26] and is necessary for attention and detection tasks. Bottom-up segmentation, where image pixels are grouped, has been studied extensively in Computer Vision, but is by nature under-determined. We instead employ a top-down segmentation scheme guided by detection, where the method "knows what it is segmenting".

Borenstein and Ullman [27] propose a class-specific top-down segmentation which provides reliable results. They learn image fragments containing class and figure-ground information from training data, and then match these fragments to a test image. The patches together form a pixelwise foreground probability map which can be used to segment objects. Combining the top-down method [27] with bottom-up segmentation, the method of Borenstein et al. [28] is able to combine the robustness of a top-down method with the local detail of bottom-up segmentation.

Bergström and Kragic [29] present an active 3D scene segmentation of unknown objects, as a basis for robot manipulation. Using the assumption that objects are placed on flat surfaces, an image is segmented into three parts, object, surface and background. An extension [30] segments several

objects in a scene. However, one can argue that detecting the object class also is important for robot manipulation.

III. Interleaved Segmentation and Detection

We base our work on the voting-based approach to object detection by interleaved segmentation and classification proposed by Leibe et al. [6]. The approach is briefly described here, see the reference for a more extensive description.

Objects are represented as a collection of visual words. The codebook of visual words for a certain object class is learned by clustering of SIFT [31] descriptors extracted from training images of the object class.

Based on the codebook, an *Implicit Shape Model* (ISM) is learned, encoding where on the object different features normally occur. For each visual word in the codebook, a distribution is learned over the spatial occurrence parameters

$$O_i^{ISM} = (x_i, y_i, s_i) \tag{1}$$

where (x_i, y_i) is the vertical and horizontal position relative to the object center and s_i the SIFT feature scale. Furthermore, each visual word is associated with a segmentation mask over the support area of the word feature.

When features are extracted from a new object image, each extracted feature is compared with the codebook entries. If the similarity to a word is above a certain threshold, it is allowed to vote for the most likely object center positions in the voting space

$$V_i^{ISM} = (x_f - x_i\frac{s_f}{s_i}, y_f - y_i\frac{s_f}{s_i}, \frac{s_f}{s_i}) \tag{2}$$

where (x_f, y_f, s_f) are the position and scale of the extracted feature. The voting uses the learned distribution over O_i^{ISM}.

Mean Shift search is then applied to find the local maxima in the voting space, i.e., the most probable object centers. Figure 2 left shows a voting example.

The mutual confidence between the features and the object hypotheses is used to define a pixel-wise figure-ground segmentation. In a top-down manner, all features that contributed to the hypothesis are backprojected and their masks combined into foreground and background probability maps, as in Figure 2 center. A figure-ground segmentation, Figure 2 right, can be obtained from the ratio of these maps.

IV. Multi-Class Detection

The ISM model of Leibe et al. [6] is designed to detect instances of a single class. To detect objects belonging to n different classes, n instances of the single class model can be trained with separate datasets of the different classes, and applied to the image independently of each other.

However, the approach with separate detectors is not scalable, as the computation time is linearly dependent on the number of classes. A realistic robotic application would involve from 20 to several thousands of object categories – rendering this approach is computationally infeasible.

The approach that we propose here is to learn a joint model with a shared codebook and use the same voting space which contain a class dimention. Such a model can take advantage of the fact that similar features are present in different classes.

For large numbers of classes n, the codebooks of different individual class detectors are likely contain very similar features, corresponding to often-occurring patterns such as straight lines, corners, etc. With a joint codebook, these features could be shared between different classes, increasing computational efficiency; the codebook size is the key issue for computation cost since the detection is based on matching codebook entries. The main advantage with a model with shared codebook is thus the decrease in computational complexity, since more and more features are shared as n grows.

Interest points are extracted from images of all n classes, as in the single class method. The codebook is then learned from clustering all features together using RNN clustering.

As we shall see in Figure 4, the gain in codebook size (i.e., the ratio of features shared between different classes) is quite moderate when performing "raw" RNN on the joint feature set. However, the codebook size can be further decreased. Given the thesis that many features are in fact shared among classes, a reasonable assumption is that many of the words in the codebook are present on instances of many of the n classes – and also for other types of objects, or in the background of images. Hence, they are not very descriptive of a certain class, or not even of the foreground areas in general. The detection performance will then not be affected if these words are removed.

The principled approach to removing uninformative codebook entries is to measure the mutual information between words and class labels [21], [32]. The *information gain* G of a certain word w_i is equal to the average pointwise mutual information between the word and all class labels,

$$G(w_i) = \frac{1}{n}\sum_{c=1}^{n} P(w_i, c)\log\frac{P(w_i, c)}{P(w_i)P(c)} \ . \tag{3}$$

Words with a high gain $G(w_i)$ are specific to a certain class, and thus correspond to unusual patterns with high discriminative power.

A principled way of decreasing codebook size is thus to remove words with an information gain lower than a certain threshold. The effects on performance of different information gain thresholds are evaluated in Section VI-A.

Based on the joint codebook, a *Joint Implicit Shape Model* (JISM) is learned, encoding on which objects, and where on the object different features are likely occur. A class parameter is added to the spatial occurrence distribution of codebook entries; for each visual word in the joint codebook, a distribution is learned over the spatial occurrences

$$O_i^{JISM} = (x_i, y_i, s_i, c_i) \tag{4}$$

where (x_i, y_i) is the vertical and horizontal position relative to the object center, s_i the SIFT feature scale, and c_i the class of the object.

As in the original method in Section III, the first step in object detection is to extract features in the image, and match

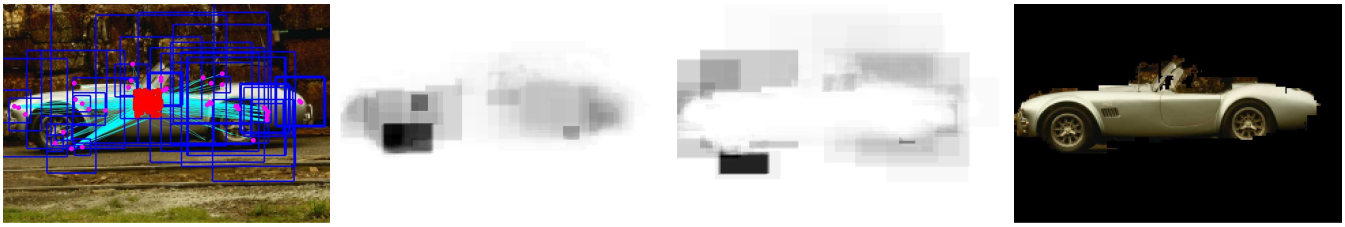


Fig. 2. Detection example. The first image shows the matched features and voting back-projected from the hypothesis; the second image shows the object probability map; the third image shows the background probability map, computed on the matched feature areas. The last image shows the segmentation from the likelihood map.

these to the joint codebook. Using the learned distributions over O_i^{JISM}, the features vote for object center positions, scales and classes in the voting space

$$V_i^{JISM} = (x_f - x_i\frac{s_f}{s_i}, y_f - y_i\frac{s_f}{s_i}, \frac{s_f}{s_i}, c_i) \quad (5)$$

in the same manner as in the original method.

Moreover, only patches belonging to the object class corresponding to the majority vote are employed for segmentation – others are considered as false positives.

V. 3D Detection

The JISM model described above suffers from different forms of noise as ISM, e.g., spurious feature detections in the background, or erroneous association of features from two different objects (for example, see Figure 8(a,d), where the space between the front wheels of the left car and the rear wheels of the right car is wrongly detected as a car).

This is addressed by using depth. Provided measurements of depth for every extracted feature, e.g., from RGB-D imagery captured with a Kinect® sensor, depth can be included among the spatial occurrence and class parameters, giving us an *Joint Implicit 3D Shape Model* (JI3SM).

The assumption underlying the JI3SM is that only features that are on similar depth can vote for the same object position. This means that depth is only explicitly involved in the detection phase, not in training. At training time, distributions over $O_i^{JI3SM} = O_i^{JISM} = (x_i, y_i, s_i, c_i)$ are learned as above.

(It would be possible to relax the assumption about similar feature depth by learning feature depths d_i relative to the object center, and add these to the measured absolute feature depth d_f in the detection stage. This would require a depth map associated with each training object instance.)

As described above, features are extracted and matched to the codebook. However, every feature is assigned a depth marking the feature location in 3D space.

The depth and scale parameters of features are (inversely) correlated, but contain slightly different information. In the light of this, we suggest the following three alternatives for augmenting the voting space.

$$\text{Alt 1}: \quad V_i^{JI3SM\ 1} = (x_f - x_i\frac{s_f}{s_i}, y_f - y_i\frac{s_f}{s_i}, \frac{s_f}{s_i}, d_f, c_i) \quad (6)$$

$$\text{Alt 2}: \quad V_i^{JI3SM\ 2} = (x_f - x_i\frac{s_f}{s_i}, y_f - y_i\frac{s_f}{s_i}, d_f, c_i) \quad (7)$$

$$\text{Alt 3}: \quad V_i^{JI3SM\ 3} = (x_f - x_i\frac{s_f}{s_i}, y_f - y_i\frac{s_f}{s_i}, d_f\frac{s_f}{s_i}, c_i) \quad (8)$$

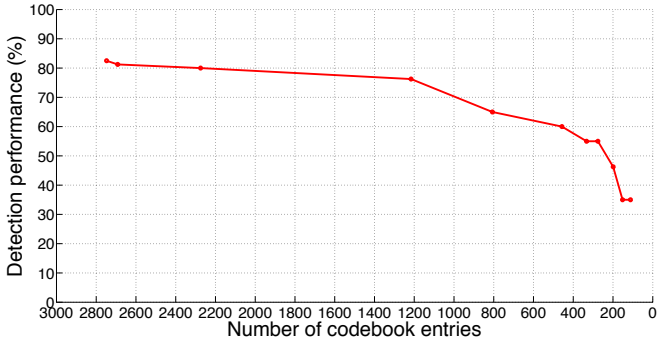


Fig. 5. Classification performance, mean classification accuracy for different parameter settings. T_{IG} is the threshold on information gain, and varies (from left to right) from 0 to 0.01. T_Q=2.

where (x_f, y_f, s_f, d_f) are the position, scale and absolute distance to the camera of the extracted feature.

The depth cue is expected to increase the robustness and accuracy of detection, since incorrect object hypotheses are less likely to appear; spurious features that accidentally support an incorrect object hypothesis rarely lie on the same depth range, On the other hand, correct features lie in the same depth range (given that the similar depth assumption holds), and are allowed to support each others votes.

The segmentation step does not currently involve depth information. A focus of future research is to explore depth information for segmentation; there is of course a high correlation between object boundaries and depth boundaries. Depth boundaries are characterized by empty areas (see Figure 8(d)), which create easily detectable ”halos” around objects. This is further discussed in the Conclusions.

VI. Results

The JISM and JI3SM were implemented in C++ on a regular desktop machine.

A. Multi-Class Detection

The JISM described in Section IV was evaluated using a four-class dataset of objects in natural outdoor settings. No depth was available due to the limitation of the Kinect® sensor to indoor scenes.

The training set consisted of 30 images of each of the classes `car side`, `cow side`, `car rear` and `motorbike`, and sampled from the dataset `http://www.vision.ee.ethz.ch/~bleibe/data`

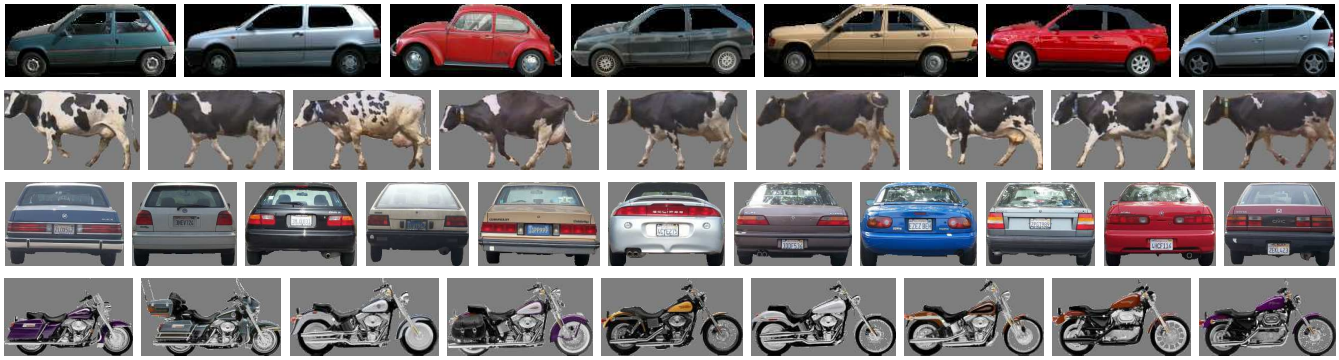
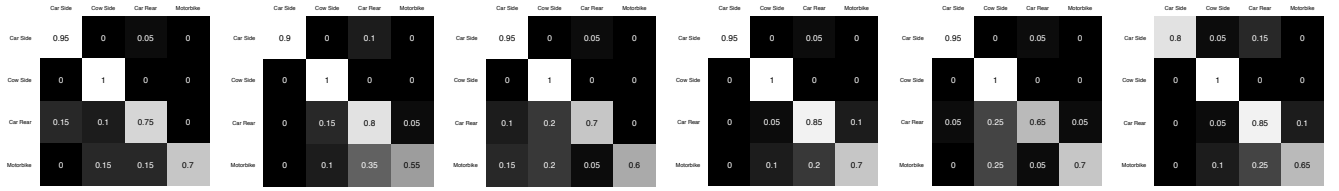


Fig. 3. Examples from the multi-class training dataset [1].



(a) Single class detectors, accuracy 85%, T_Q=0, 7825 words (b) Multi-class detector, accuracy 81%, T_Q=0, 6949 words (c) Single class detectors, accuracy 81%, T_Q=1, 4777 words (d) Multi-class detector, accuracy 88%, T_Q=1, 4578 words (e) Single class detectors, accuracy 83%, T_Q=2, 2748 words (f) Multi-class detector, accuracy 83%, T_Q=2, 2680 words

Fig. 4. Classification performance, confusion matrices for different parameter settings. T_Q is the threshold on codebook cluster size.

`/datasets.html`, see Figure 3. All images were labeled with a segmentation mask. For testing, 20 unseen images (not in the training set) of the three first classes were chosen randomly from the same dataset. To increase the difficulty of the `motorbike` test cases, 20 single motorbikes images were chosen randomly from the TUD dataset `http://www.mis.tu-darmstadt.de/datasets`.

The JISM was trained using the training set above. For comparison, four individual ISM instances were also trained with each of the four classes of the training set. The Harris-Laplace detector and SIFT descriptor from [33] were used for feature extraction.

In this experiment, only the classification aspect of the detection was evaluated. All four ISMs were applied to each image in the test set, and the object hypothesis As exawith highest score was regarded as the classification result of that image. Figure 4(a) shows the classification result with the multiple single class models and Figure 4(b) shows the classification result with the multi-class model. We can see that the same level of detection accuracy can be achieved with smaller codebook size. The single class model codebooks contain 7825 entries together, while the multi-class codebook size is only 6949. The reason for the 10% decrease in codebook size is that words are shared between classes. It should be noted that the number of classes is quite small – more classes should lead to a larger decrease. Further, we will discuss that with multi-class detection, many existing codebook selection method can be easily applied.

It is argued [6] that it is safe to ignore clusters with only one feature, as they likely correspond to class outliers. Figure 4(c) shows the detection result with the multiple single class models and codebooks with clusters containing > 1 feature, and Figure 4(d) shows the detection result with the multi-class model and the joint codebook with clusters containing > 1 feature. The multi-class model performed slightly better than the single class model. However, the decrease in codebook size was smaller with thresholding, since one-feature clusters are less likely with a larger set of features to cluster – removing clusters with 1 feature in the single class case might be comparable to removing multi-class clusters with ≤ 2 features. Figures 4(e) and 4(f) show the result of removing clusters of size ≤ 2.

As discussed in Section IV, the size of the joint codebook can be decreased further by removing words with a low information gain. Figure 5 shows the average classification performance as a function of the codebook size, which in turn depends on the threshold on information gain. We can see from the figure that with information gain as a codebook selection criterion, the codebook size can be decreased more than 50% with less than 5% of classification accuracy loss.

B. 3D Detection

We then evaluated the addition of the depth cue, described in Section V. We assume that the changes in performance with and without depth are independent of the changes in performance with and without multi-class detection; a reasonable assumption given that the changes in data representation are themselves uncorrelated. Given this assumption, it is sufficient to evaluate the depth cue addition using training data containing a single class ($c_i = 1$).

A training dataset of side views of 40 different toy cars was collected with a Kinect® sensor (see Figure 6). Due to sensor limitations, an indoor setting was used. The variation in physical size was high in the dataset – higher than among

Fig. 6. Examples from the 3D toy car dataset.

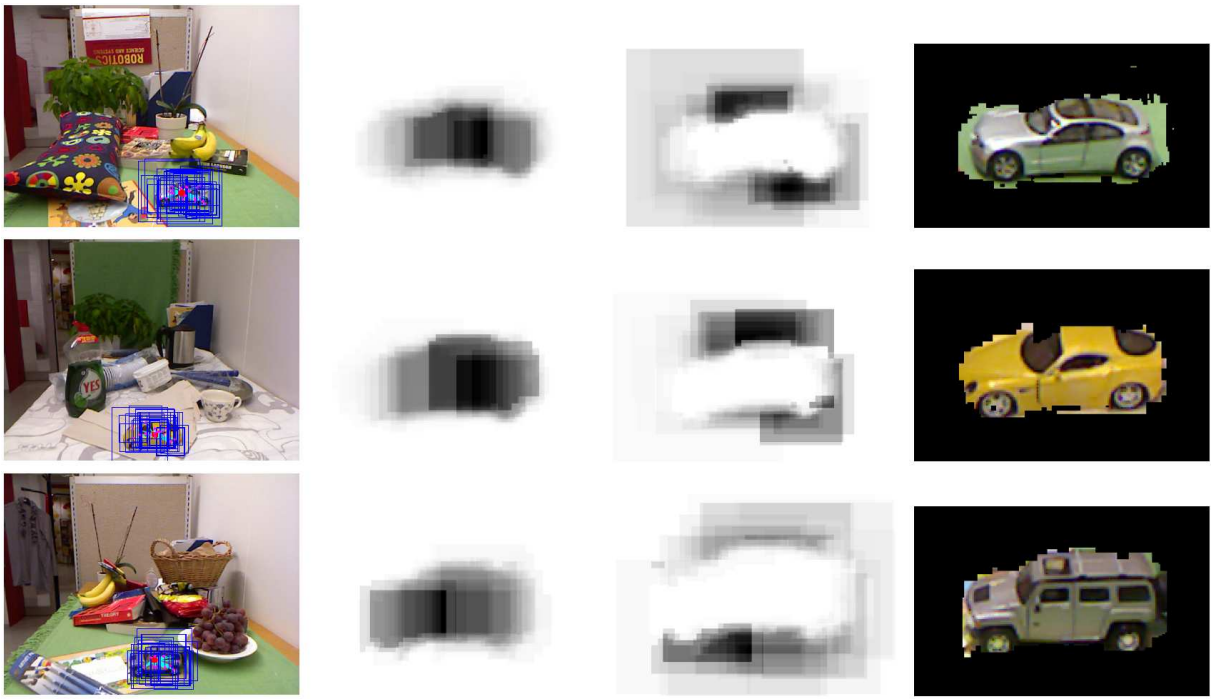


Fig. 7. Detection examples using $V^{JI3SM\ 3}$ (Eq (8)). For every row, the first image shows the matched features and voting back projected from the hypothesis; the second image shows the object probability map; the third image shows the background probability map, computed on the matched feature areas; the last image shows the segmentation from the likelihood map. To lower the number of detections and the size of the detection for illustrative purposes, a smaller bandwidth was used than in Table 1: 5 for x and y, and 0.07 for $s \times d$.

real cars. A version of the training set was therefore prepared, where all training images were scaled to the same width. (The test images were kept unscaled.)

For the first experiment below, a set of test images were collected, where cars from the training set were placed in different, cluttered environments, one car per image (see Figure 7 left). The test images are very challenging for three reasons: Firstly, the cars occupy a very small part of the images, secondly, the background is highly textured, and last, the cars vary in scale with more than a factor 2.

No car instance was present in both the training and test sets in the same run. Five training datasets were generated by randomly selecting training images of 30 toy cars for training, and cluttered test images of the remaining 10 unseen toy cars for testing. All experiments were then performed independently five times on the different datasets.

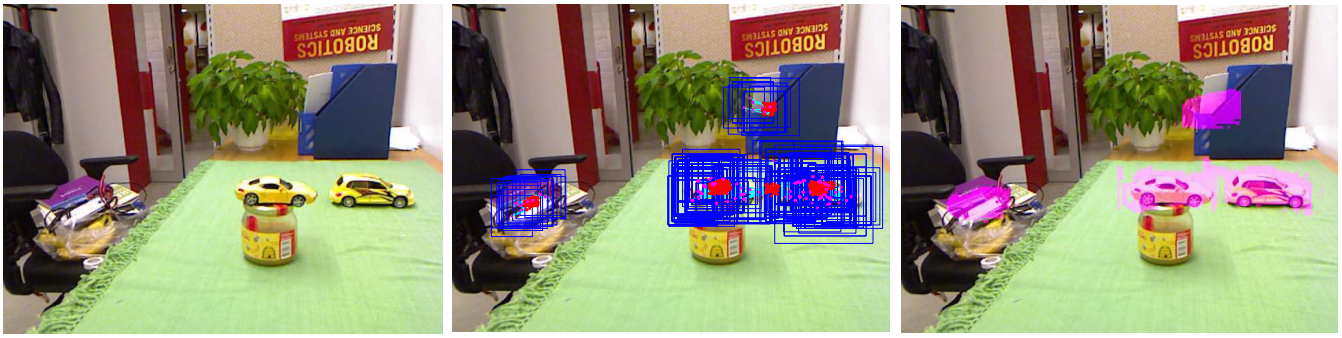
A hypothesis brought forward in Section V is that the depth cue increases the robustness to background noise. One way to measure this is to study the voting confidence, i.e., to what degree votes agree on the same object hypothesis. The results were:
With V^{ISM} (Eq (2)): 58%,
With $V^{JI3SM\ 1}$ (Eq (6)): 62%,
With $V^{JI3SM\ 2}$ (Eq (7)): 64%,
With $V^{JI3SM\ 3}$ (Eq (8)): 66%.
For voting, a bandwidth of 10 for x and y, 0.01 for s, and 0.15 for d and $s \times d$ was used. We conclude that the voting confidence is indeed higher with depth cues, and that the $V^{JI3SM\ 3}$ voting space gives the highest confidence.

The detection performance was then evaluated in terms of precision and recall. For this experiment, another test dataset was collected. Four car instances were randomly selected from the 40 cars, and the corresponding training images were removed from the training set (Figure 6). A series of 32 images with two or more test instances were then collected; for an example, see Figure 8(a,d).

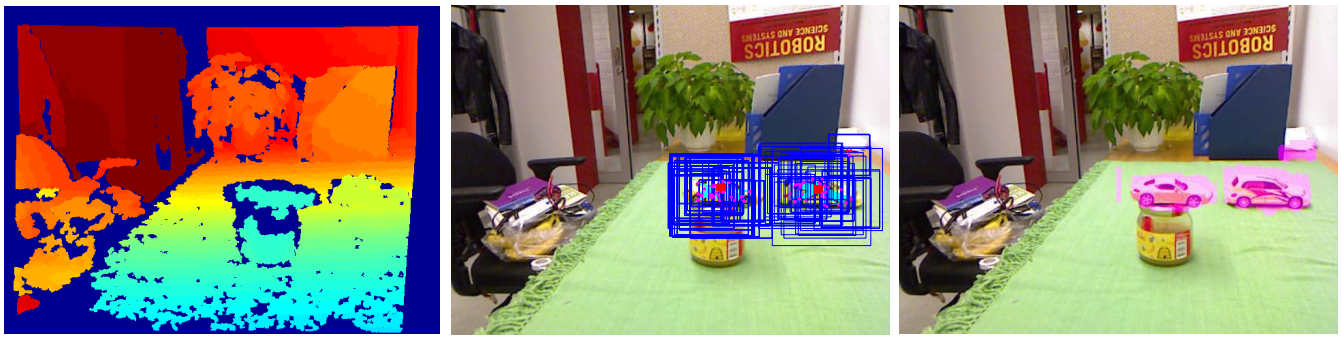
Detection in the images was carried out in the following way: Features were extracted, which voted for car hypotheses. Object hypotheses, i.e., local maxima in the hypothesis confidence space, were then detected. All hypotheses with a probability higher than T_{ratio}=55% of the probability of the strongest hypothesis were maintained as detections.

Figure 8 shows an example detection result from this dataset; (e) showing the detection and (f) the segmentation of $V^{JI3SM\ 3}$, and as a baseline, (b) showing the detection and (c) the segmentation of V^{ISM} which is not using depth.

The 2D detection in Figure 8(b) shows examples of the kind of erroneous detections that might occur when depth



(b) 2D detection, baseline (c) 2D segmentation, baseline



(d) Depth image (e) 3D detection (f) 3D segmentation

Fig. 8. Example of multi-object detection. (b) and (c) show detection and segmentation using V^{ISM} which does not employ depth information – two correct detections and three false detections. (e) and (f) show the detection and segmentation using $V^{JI3SM\ 3}$ – two correct detections and one false detection. A smaller bandwidth was used than in Figure 9: 5 for x and y, and 0.07 for $s \times d$

is not used. First of all, enough car-like spurious features were found on the plant, the blue folder and the chair, to give rise to two false detections. The features for both these detections are on different depths, which explains why they do not vote for the same hypothesis in the 3D detection in Figure 8(e). Secondly, a more systematic error occurs: the two cars are aligned in the image so that the rear wheel of the right car supports the same hypothesis as the front wheel of the left car – a car is "hallucinated" between the two real cars. However, since the cars are on significantly different depth (see Figure 8(d)), this false hypothesis does not occur if depth is taken into account (see Figure 8(e)).

Figure 9 gives the precision-recall curves of the first stage detection result using the four alternative voting spaces. The curves were generated by varying the threshold (T_{ratio}) of the ratio between the probability of detections and that of the strongest detection in the image. This threshold controls the recall factor as: $(T_{\text{ratio}} \to 0) \Leftrightarrow (\text{recall} \to 1)$.

The curves confirm that using depth in the detection gives a more stable performance than detection with 2D cues only. However, increasing the dimensionality of the voting space, as in $V^{JI3SM\ 1}$ (green curve) gives a significantly worse detection performance. The reason for this is most certainly that the JI3SM model with the larger space requires more training data – the current training set of 36 cars is simply too small for the model to converge. Future work includes evaluation with a larger training set (Section VII).

VII. Conclusions

We presented a 3D, multi-class object detection and segmentation method, intended for a humanoid robot perception application. To that end, we extended the 2D, single-class detection method of Leibe et al. [6] to handle multiple classes using a joint minimal codebook, and to incorporate depth measurements to enhance the robustness of the voting procedure of the detection step. Both these extensions are essential for the method to be of use on a humnoid robot functioning in human environments. Moreover, we still kept all the advantages of the original method, e.g., rough pose estimation by learning classes corresponding to both viewing direction and object class.

The experiments showed that with the new multi-class model, the same detection accuracy could be obtained as with a set of single-class models, but with a gain in codebook size: the codebook size could be lowered to half with less than 5% detection accuracy loss. Moreover, it was shown that the introduction of a depth cue in the method improved detection performance, in that votes from spurious background features and other objects were filtered out more efficiently in the object detection stage.

As discussed in Section V, depth is not currently used for segmentation explicitly. However, there is rich information in the range image which could be exploited for that purpose. Using shape features [34], [35] together with visual features (e.g., SIFTs) would most certainly increase both the detection and 3D segmentation performance.

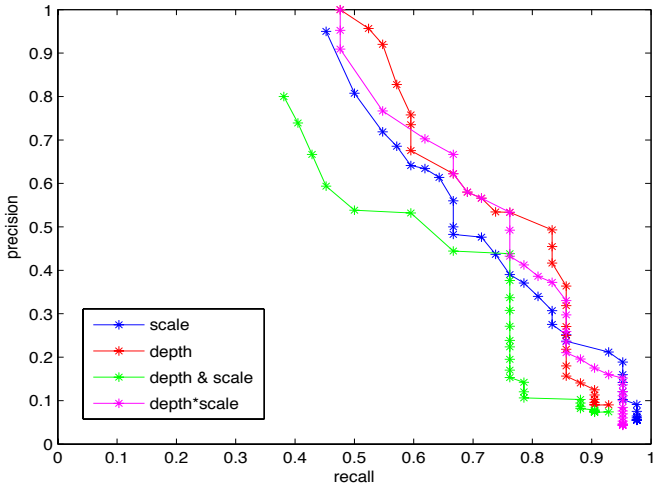


Fig. 9. Precision-recall curves for detection using the V^{ISM} (blue), $V^{JI3SM\ 1}$ (green), $V^{JI3SM\ 2}$ (red), and $V^{JI3SM\ 3}$ (magenta) voting spaces.

We also intend to integrate the current method on a real humanoid robot, and further evaluate the performance of 3D detection and segmentation of different classes of indoor objects. This involves collecting a database of RGB-D images of a large number of object classes, including several instances of each class.

Another avenue of research to explore further is that of more elaborate category models. When the number of classes grow, the classification is improved by introducing structure, e.g., topic models such as pLSA [36], LDA [37] or DiscLDA [38]. Such hierarchical and structural models are also more suited for reasoning about objects in grasping and manipulation applications, where topics or features can be correlated to robot grasping strategies [39].

REFERENCES

[1] A. Torralba, K. Murphy, W. T. Freeman, and M. A. Rubin, "Context-based vision system for place and object recognition," in *ICCV*, 2003.

[2] A. Pronobis, O. M. Mozos, B. Caputo, and P. Jensfelt, "Multi-modal semantic place classification," *IJRR*, vol. 29, no. 2-3, pp. 298–320, 2010.

[3] A. Aydemir, K. Sjöö, J. Folkesson, A. Pronobis, and P. Jensfelt, "Search in the real world: Active visual object search based on spatial relations," in *ICRA*, 2011.

[4] P. E. Forssén, D. Meger, K. Lai, S. Helmer, J. J. Little, and D. G. Lowe, "Informed visual search: Combining attention and object recognition," in *ICRA*, 2008.

[5] Z.-C. Marton, D. Pangercic, N. Blodow, and M. Beetz, "Combined 2d-3d categorization and classification for multimodal perception systems," *Int. J. Rob. Res.*, vol. 30, pp. 1378–1402, September 2011. [Online]. Available: http://dx.doi.org/10.1177/0278364911415897

[6] B. Leibe, A. Leonardis, and B. Schiele, "Robust object detection with interleaved categorization and segmentation," *IJCV*, vol. 77, no. 1-3, pp. 259–289, 2008.

[7] J. Bohg, M. Johnson-Roberson, B. Léon, J. Felip, X. Gratal, N. Bergström, D. Kragic, and A. Morales, "Mind the gap – robotic grasping under incomplete observation," in *ICRA*, 2011.

[8] Z. Marton, L. Goron, R. Bogdan Rusu, and M. Beetz, "Reconstruction and verification of 3D object models for grasping," in *Robotics Research, STAR 70*, C. Pradalier, R. Siegwart, and G. Hirzinger, Eds. Springer, 2011, pp. 315–328.

[9] M. Björkman and D. Kragic, "Active 3D scene segmentation and detection of unknown objects," in *ICRA*, 2010.

[10] M. Piccardi, "Background subtraction techniques: a review," in *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, vol. 4, oct. 2004, pp. 3099 – 3104 vol.4.

[11] P. Viola and M. J. Jones, "Robust real-time object detection," in *IEEE Workshop on Statistical and Computational Theories of Vision*, 2001.

[12] B. Wu and R. Nevatia, "Improving part based object detection by unsupervised, online boosting," in *CVPR*, 2007.

[13] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman, "Multiple kernels for object detection," in *ICCV*, 2009.

[14] A. Opelt, A. Pinz, and A. Zisserman, "A boundary-fragment-model for object detection," in *ACCV*, 2006.

[15] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *PAMI*, vol. 32, pp. 1627–1645, 2010.

[16] P. F. Felzenszwalb, R. B. Girshick, and D. McAllester, "Cascade object detection with deformable part models," in *CVPR*, 2010.

[17] M. Sun, G. Bradski, B.-X. Xu, and S. Savarese, "Depth-encoded hough voting for joint object detection and shape recovery," in *Computer Vision ECCV 2010*, ser. Lecture Notes in Computer Science, K. Daniilidis, P. Maragos, and N. Paragios, Eds., 2010, vol. 6315, pp. 658–671.

[18] N. Razavi, J. Gall, and L. Van Gool, "Scalable multi-class object detection," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, june 2011, pp. 1505 –1512.

[19] J. Shotton, A. Blake, and R. Cipolla, "Contour-based learning for object detection," in *ICCV*, 2005.

[20] V. Ferrari, L. Fevrier, F. Jurie, and C. Schmid, "Groups of adjacent contour segments for object detection," *PAMI*, vol. 30, no. 1, pp. 36–51, 2008.

[21] J. Yang, Y.-G. Jiang, A. G. Hauptmann, and C.-W. Ngo, "Evaluating bag-of-visual-words representations in scene classification," in *Workshop on Multimedia Information Retrieval*, 2007.

[22] A. Torralba, K. P. Murphy, and W. T. Freeman, "Sharing visual features for multiclass and multiview object detection," *PAMI*, vol. 29, pp. 854–869, 2007.

[23] L. Liu, L. Wang, and C. Shen, "A generalized probabilistic framework for compact codebook creation," in *CVPR*, 2011.

[24] K. Lai, L. Bo, X. Ren, and D. Fox, "A large-scale hierarchical multi-view RGB-D object dataset," in *ICRA*, 2011.

[25] ——, "Sparse distance learning for object recognition combining RGB and depth information," in *ICRA*, 2011.

[26] N. R. Pal and S. K. Pal, "A review on image segmentation techniques," *Pattern Recognition*, vol. 26, no. 9, pp. 1277–1294, 1993.

[27] E. Borenstein and S. Ullman, "Class-specific, top-down segmentation," in *ACCV*, 2002.

[28] E. Borenstein, E. Sharon, and S. Ullman, "Combining top-down and bottom-up segmentation," in *Computer Vision and Pattern Recognition Workshop*, 2004.

[29] N. Bergström, M. Björkman, and D. Kragic, "Generating object hypotheses in natural scenes through human-robot interaction," in *Intelligent Robots and Systems, 2011. IROS 2011. IEEE/RSJ International Conference on*, sept. 2011, pp. 827–833.

[30] M. Johnson-Roberson, J. Bohg, M. Björkman, and D. Kragic, "Attention-based active 3D point cloud segmentation," in *IROS*, 2010.

[31] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, vol. 60, no. 2, pp. 91–110, 2004.

[32] Y. Yang and J. O. Pedersen, "A comparative study on feature selection in text categorization," in *ICML*, 1997.

[33] K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek, "Evaluating color descriptors for object and scene recognition," *PAMI*, vol. 32, pp. 1582–1596, 2010.

[34] A. E. Johnson and M. Hebert, "Using spin images for efficient object recognition in cluttered 3D scenes," *PAMI*, vol. 21, pp. 433–449, 1999.

[35] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (FPFH) for 3D registration," in *ICRA*, 2009.

[36] A. Bosch, A. Zisserman, and X. Munoz, "Scene classification via pLSA," in *ACCV*, 2006.

[37] L. L. Fei-Fei and P. Perona, "A bayesian hierarchical model for learning natural scene categories," in *CVPR*, 2005.

[38] Z. Niu, G. Hua, X. Gao, and Q. Tian, "Spatial-discLDA for visual recognition," in *CVPR*, 2011.

[39] D. Song, K. Huebner, V. Kyrki, and D. Kragic, "Learning task constraints for robot grasping using graphical models," in *IROS*, 2010.